NASA TECH BRIEF
Ames Research Center

NASA Tech Briefs announce new technology derived from the U.S. space program. They are issued to encourage commercial application. Tech Briefs are available on a subscription basis from the National Technical Information Service, Springfield, Virginia 22151. Requests for individual copies or questions relating to the Tech Brief program may be directed to the Technology Utilization Office, NASA, Code KT, Washington, D.C. 20546.

Graphical Method for Analyzing Digital Computer Efficiency

The problem:

To optimize the computational efficiency of general-purpose digital computers, taking into account such factors as computation costs as measured by computing time, hardware requirements, and error probabilities.

The solution:

An analysis method utilizing the graph-theoretic approach for evaluating the computation cost. This method is new in that it makes a logical distinction between the linear graph of a computation and the linear graph of a program and shows the importance of the relationships between these two graph models without going into any great depth to describe the algebra of the relationships. Because the method is general, it applies equally well to schedules, signal flow graphs, and any other process which depends on quantitative edge nomenclature and precedence relationships between edges.

How it's done:

The first step in the analysis of a computational task is the generation of a linear oriented graph, known also as a directed graph or digraph, representing the computation process. A quantitative evaluation is assigned to the nodes or vertices of the graph, and a precedence relationship, i.e., the essential topology of the graph, is assigned to the directed edges. In a sequence of operations within a computational task, a vertex represents a significant temporal point or the instant that a particular operation begins or ends. Since the graph corresponds closely to a signal flow graph and the type of graph used in communications theory, it benefits from the mathematical developments of these related fields.

A typical simplified model of the linear oriented graph of a computer task (solution of the pythagorean formula) is given in Figure 1. In this example, as in every computational task, certain precedence relationships remain inviolate. For example, for this function, the squares must be taken before summing and each sum must be completed before the final square root operation is performed. This simplified linear oriented computation graph is topologically very different from the graph in Figure 2, which represents a simple sequential computer program for this function. A set of rules or an algorithm could be developed for systematically generating the program graph from the computation graph. For the more complex programming tasks, where a multiprocessing capability exists, the program graph for the simultaneous use of two processors would be topologically identical to the computation graph. As a result, the computation task would take less time.

In the topological analysis of a computation task, the following terms are essential:

(a) Edges represent computing operations, and the weight or value assigned to an edge represents the set of quantities that measure the computing operation, such as probability of execution, execution time, memory requirement, chance of error, etc.
(b) Vertices represent the beginning and end of an operation.
(c) Paths represent a sequence of operations, with weights or values computed as joint probabilities, total time of execution, and total chance of error, etc., for the edges traversed.
(d) Loops (recursive or indexed) represent cyclic operations or a recurring set of identical operations.

(continued overleaf)

This document was prepared under the sponsorship of the National Aeronautics and Space Administration. Neither the United States Government nor any person acting on behalf of the United States Government assumes any liability resulting from the use of the information contained in this document, or warrants that such use will be free from privately owned rights.

In order to evaluate quantitatively a computation graph containing a deterministic cycle, quantitative information such as operation time and probability of occurrence is assigned for each edge of the graph, and all cyclic operations or loops are identified and transformed into acyclic equivalents. A program graph can then be developed on the basis of some fixed computer hardware configuration. This program graph produces the desired results in the form of the computation task cost (measured in seconds) and the output occurrence probability. When the probability of occurrence of each edge is unity, the output is a certainty.

If the hardware can be varied to provide the greatest amount of computing parallelism, the shortest possible computation time is proportional to the longest path of the computation graph from start to finish, whereas the longest time is the sum of all operation times.

The computation graph for computing a probabilistic function becomes more complex, and the function program graph using serial computing hardware is not a single path, but two alternate paths. The probability of output is the sum of the joint probabilities through each path.

Note:

Requests for further information may be directed to:

Technology Utilization Officer
Ames Research Center
Moffett Field, California 94035
Reference: TSP71-10453

Patent status:

No patent action is contemplated by NASA.

Source: R.M. Munoz
Ames Research Center
and S. Park Chan of
University of Santa Clara
under grant from
Ames Research Center
(ARC-10210)